

AMENDMENTS TO THE CLAIMS

The following listing of claims replaces all prior versions, and listings, of claims in the captioned patent application:

1. (Currently Amended) In a J2EE application server, a method for monitoring performance of a plurality of transactions including a top level transaction and plurality of transactions relating to said top level transaction in a child parent hierarchy, comprising, for each of selected ones of said plurality of transactions, obtaining a performance metric corresponding to the selected transaction by:

installing an instrument hook ~~upon loading prior to execution of~~ the selected transaction; ~~and~~

instrumenting said selected transaction upon execution of the selected transaction using one or more plug-in instruments called by the instrument hook[[.]]; ~~and~~

initiating said top level transaction in response to a request received from a web server; ~~for each of said instrumented transactions,~~

transmitting a cookie from said web server to said application server together with said request;

generating a correlator correlators for identifying said top level transaction and a parent transaction, if any, of said instrumented transaction upon execution of each of said instrumented transactions; ~~and~~

utilizing said cookie for the correlator identifying said top level transaction; and

utilizing said correlators to cross-correlate a performance metric corresponding to a parent transaction with one or more performance metrics corresponding to one or more child transactions of said parent transaction,

wherein the one or more plug-instruments implement an interface that communicates data for the performance metric.

2. (Previously Presented) The method of claim 1, wherein the step of instrumenting said selected transaction comprises inserting instrumentation code in a bytecode representation of said selected transaction.

3. (Previously Presented) The method of claim 2, wherein said performance metric corresponds to a response time of said selected transaction.
4. (Previously Presented) The method of claim 3, wherein said instrumentation code effects generation of a start time marker upon start of execution of said selected transaction and generation of a stop time marker upon completion of execution of said selected transaction.
5. (Original) The method of claim 4, wherein said instrumentation code generates calls to an Application Response Measurement (ARM) agent to cause generation of said stop and start time markers.
6. (Previously Presented) The method of claim 5, further comprising utilizing said start and stop time markers to measure a response time of said selected transaction.
7. (Previously Presented) The method of claim 1, further comprising generating a record for each instrumented transaction upon completion of said instrumented transaction, said record indicating said performance metric associated with said instrumented transaction, a parent of said instrumented transaction, and said top level transaction.
8. (Previously Presented) The method of claim 7, further comprising transmitting said instrumented transaction record to an analysis and presentation module.
9. (Original) The method of claim 1, further comprising storing said correlators in a thread local storage stack in case of execution of said hierarchical transactions in a single thread.
10. (Original) The method of claim 9, further comprising storing said correlators in the stack based on a LIFO protocol.
11. (Previously Presented) The method of claim 10, further comprising removing one correlator of said correlators from said stack upon completion of said hierarchical transaction associated with said correlator.

12-14. (Canceled)

15. (Currently Amended) A method for monitoring performance of at least two Java transactions that are related to one another as parent-child transactions, comprising obtaining a performance metric corresponding to each of said at least two Java transactions by:

installing an instrument hook upon loading prior to execution of each of said at least two Java transactions, wherein a top level transaction of the at least two Java transactions is initiated in response to a request received from a web server and said web server transmits a cookie to an application server together with said request; and

instrumenting each of said at least two Java transactions upon execution of each of said at least two Java transactions using one or more plug-in instruments called by the instrument hook;

generating a correlator corresponding to said parent transaction, utilizing RMI over IIOP to send said parent correlator incorporated in a header of an IIOP message to said child transaction upon execution; and

generating another correlator corresponding to said child transaction; and
generating an additional correlator corresponding to the top level transaction
utilizing said cookie, and

wherein the one-or more plug-instruments implement an interface that
communicates data for the performance metric.

16. (Previously Presented) The method of claim 15, further comprising employing said correlators to cross correlate the performance metric of said parent transaction with the performance metric of said child transaction.

17. (Previously Presented) The method of claim 15, wherein said performance metric corresponds to a response time of each of said at least two Java transactions.

18. (Previously Presented) The method of claim 17, wherein said performance metric corresponds to a start time marker upon start of execution of each of said at least two Java transactions and a stop time marker upon completion of execution of each of said at least two Java transactions.

19. (Original) The method of claim 18, wherein said modified bytecode representation generate calls to an Application Response Measurement (ARM) agent to cause generation of said start and stop time markers.

20. (Currently Amended) A computer readable medium comprising instructions operable by a computer which when executed causes the computer to perform a method comprising:

obtaining a performance metric corresponding to a selected transaction of a plurality of parent-child transactions by

installing an instrument hook upon loading the selected transaction; and

instrumenting said selected transaction upon execution of the selected

transaction using one or more plug-in instruments called by the instrument hook; and

generating correlators for each of said transactions, wherein each correlator identifies said a top level transaction and a parent transaction, if any, corresponding to its associated transaction, wherein said top level transaction is initiated in response to a request received from a web server, and wherein said web server transmits a cookie to an application server together with said request,

utilizing said cookie to generate said correlator for said top level transaction, and

wherein the one-or more plug-instruments implement an interface that communicates data for the performance metric.